

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

SIMULOVÁNÍ MULTICASTOVÝCH PŘENOSŮ
V SIMULÁTORU OMNET++

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PETR MATELEŠKO

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

SIMULOVÁNÍ MULTICASTOVÝCH PŘENOSŮ V SIMULÁTORU OMNET++

MULTICAST TRANSMISSIONS SIMULATION IN OMNET++

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PETR MATELEŠKO

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. PETR MATOUŠEK, Ph.D.

Abstrakt

Tato bakalářská práce se zabývá možnostmi simulace multicastu v simulátoru OMNeT++. Popisuje základní principy multicastového přenosu dat a podrobně se věnuje protokolu IGMP ve všech jeho verzích. Praktickou částí práce je návrh a implementace rozšiřujícího modulu protokolu IGMP do simulátoru OMNeT++.

Abstract

This bachelor thesis deals with the possibilities of multicast simulation in OMNeT++ simulator. It describes basic principles of the multicast data transfer and closely focuses on the IGMP protocol in all of its versions. The applied part of this thesis consists of a design and implementation of an IGMP extension module for the OMNeT++ simulator.

Klíčová slova

multicast, IGMP, OMNeT++, Cisco, simulace počítačové sítě

Keywords

multicast, IGMP, OMNeT++, Cisco, computer network simulation

Citace

Petr Mateleško: Simulování multicastových přenosů v simulátoru OMNeT++, bakalářská práce, Brno, FIT VUT v Brně, 2010

Simulování multicastových přenosů v simulátoru OMNeT++

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Petra Matouška, Ph.D.

.....
Petr Mateleško
19. května 2010

Poděkování

Děkuji svému vedoucímu Ing. Petrovi Matouškovi, PhD a ostatním kolegům z ANSA týmu za odborné rady, podporu a ochotu pomoci během zpracovávání práce.

© Petr Mateleško, 2010

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah.....	1
Seznam obrázků.....	3
1 Úvod.....	4
1.1 Cíl práce.....	4
1.2 Struktura práce.....	4
2 Základní principy multicastu.....	5
2.1 Dělení počítačové komunikace.....	5
2.2 Adresace.....	5
2.3 Popis multicastové komunikace.....	6
2.4 Multicastové směrování.....	7
3 IGMP.....	8
3.1 Základní pohled na IGMP.....	8
3.2 IGMP verze 1.....	9
3.2.1 Formát a typy zpráv.....	9
3.2.2 Popis chování.....	9
3.3 IGMP verze 2.....	10
3.3.1 Formát a typy zpráv.....	10
3.3.2 Proměnné a časovače.....	12
3.3.3 Popis chování.....	13
3.4 IGMP verze 3.....	16
3.4.1 Formát a typy zpráv.....	17
3.4.2 Popis chování.....	20
3.5 IGMP na Cisco zařízeních.....	21
4 OMNeT++.....	23
4.1 OMNeT++.....	23
4.2 INET framework.....	23
5 Popis implementace.....	24
5.1 Návrh modulu.....	24
5.2 Implementace modulu.....	25
5.3 Zprávy.....	25
5.4 Časovače.....	26
5.5 Rozhraní IGMP.....	27
5.6 Zpracování zpráv.....	28
6 Testovací simulace.....	29

6.1 Test pravidelného odesílání Query.....	30
6.2 Test volby dotazovače.....	30
6.3 Test přihlášení do skupiny.....	30
6.4 Test odhlášení ze skupiny.....	31
6.5 Test vypršení platnosti skupiny.....	31
7 Závěr.....	32
Literatura	33
Obsah CD.....	34
Implementace IGMP.....	35

Seznam obrázků

2.1: Protokoly používané v multicastové síti.....	8
3.1: Formát IGMP v1 paketu.....	10
3.2: Formát IGMP v2 paketu.....	12
3.3: Stavový diagram IGMP pro klienty.....	17
3.4: Stavový diagram IGMP směrovače	18
3.5: Formát zprávy Membership Query v3.....	19
3.6: Formát zprávy Membership Report v3.....	20
3.7: Formát Group Record.....	21
5.1: Umístění modulu IGMP jako součásti modulu NetworkLayer.....	26
6.1: Testovací topologie.....	31
6.2: Zobrazení tabulky IGMP rozhraní.....	31
6.3: Rozhraní je zařazeno do multicastových skupin.....	32
6.4: Stav InterfaceTable po odstranění skupiny.....	33
6.5: Stav InterfaceTable po vypršení platnosti skupiny.....	33

1 Úvod

Technologie přenosu dat pomocí multicastu má ve světě počítačových sítí velký potenciál. Je vhodné ji používat všude tam, kde jeden nebo více zdrojů vysílá svá data a větší počet odběratelů tyto data přijímá. Setkáváme se s ní zejména při multimediálních přenosech, jejichž objem vlivem nejrůznějších okolností v dnešní době stále narůstá. S praktickým nasazením a využíváním multicastu v reálných sítích vyvstává potřeba zkoumat chování sítě při nejrůznějších stavech a situacích, aniž bychom byli nuceni použít reálná síťová zařízení. Náš požadavek můžeme naplnit pomocí počítačové simulace. S využitím patřičného simulačního nástroje můžeme namodelovat síťovou komunikaci, experimentovat s ní, spouštět ji za různých okolností, sledovat zajímavé parametry a analyzovat naměřené hodnoty. Výsledky simulace pak můžeme zohlednit při návrhu či optimalizaci reálných počítačových sítí.

1.1 Cíl práce

Tato práce si klade za cíl představit multicastovou komunikaci v kontextu počítačové simulace. V praktické části se zabývá simulačním nástrojem OMNeT++ a jeho využitím při simulaci multicastu. Stěžejní částí práce je návrh a implementace rozšiřujícího modulu simulátoru OMNeT++ pro simulaci protokolu IGMP, jednoho z protokolů nezbytných pro multicastovou komunikaci.

1.2 Struktura práce

Ve druhé kapitole jsou popsány základní principy multicastu, dozvíme se jaké technologie jsou nutné pro funkční multicastovou komunikaci.

Třetí kapitola popisuje protokol IGMP ve všech jeho verzích, zvláště se zaměřením na verzi druhou. Teoretický popis protokolu nám poslouží jako podklad pro jeho praktickou implementaci. Také jsou zde uvedeny možnosti konfigurace Cisco zařízení ohledně protokolu IGMP.

Čtvrtá kapitola pojednává o simulačním nástroji OMNeT++, frameworku INET a prostředcích, které nám tyto nástroje nabízí pro simulaci multicastu.

V páté kapitole je popsána praktická implementace rozšiřujícího modulu IGMP do simulátoru OMNeT++.

Šestá kapitola se zabývá testovací simulací nově vytvořeného modulu a ověřením jeho vlastností.

V poslední kapitole je práce shrnuta, zhodnocena a je nastíněn možný další vývoj.

2 Základní principy multicastu

V této kapitole jsou uvedeny základní principy multicastu, které jsou důležité pro pochopení problematiky a potřeby simulace. Multicast je zde popsán na IPv4, variantou IPv6 se práce nezabývá. Hlouběji se multicastu věnuje například diplomová práce Ing. Vladimíra Veselého [1], kde lze načerpat podrobnější a komplexnější informace.

2.1 Dělení počítačové komunikace

V počítačových sítích dělíme komunikaci podle okruhu příjemců na *unicast*, *broadcast* a *multicast*:

Unicast – Jedná se o nejběžnější druh komunikace, existuje vždy jeden zdroj a jeden příjemce. Zdroj odesílá data pro každého příjemce zvlášť, byť by více příjemců mělo zájem o stejná data. Nevýhodou může být zbytečná duplikace paketů v případě aplikací s širokým okruhem příjemců, kteří požadují stejná data od jednoho zdroje.

Broadcast – Jedná se o opak *unicastu*, data od zdroje putují vždy všem příjemcům bez ohledu na to, zda příjemce data požadoval či ne. Nevýhodou je zatěžování sítě pakety, které příjemci nežádali a musí je zpracovat.

Multicast – Při tomto typu komunikace existuje jeden nebo více zdrojů dat a okruh příjemců, kteří si sami rozhodují, jestli budou data přijímat či ne. Příjemci jsou podle svého zájmu o data sdruženi do multicastových skupin, jedná se tedy o tzv. skupinovou komunikaci. Do těchto skupin se mohou libovolně přihlašovat a odhlašovat. Při cestě dat od zdroje k cílům nedochází k nadbytečnému zmnožení paketů, ty se až replikují v průběhu cesty na směrovačích, jen když je to opravdu nutné – příjemci leží v rozdílných sítích. Pro určitý typ aplikací je multicast na rozdíl od unicastu a broadcastu efektivnější. Multicast se typicky používá při přenosech multimediálních dat jako streaming zvuku, videa, konference mezi více účastníky, ale také při synchronizaci času. Mezi nevýhody multicastu patří nespolehlivá komunikace, data jsou přenášeny pomocí UDP s možností určité ztráty.

2.2 Adresace

Pro adresování multicastových skupin se na úrovni síťové vrstvy využívají IP adresy třídy D. Adresní rozsah je tedy 224.0.0.0 – 239.255.255.255. Multicastové adresy dělíme do 3 skupin podle jejich použití:

- **224.0.0.0 – 224.0.0.255** – Využívají je síťové protokoly jako IGMP, OSPF, RIP a další. Mají TTL = 1, tudíž se používají jen v lokální podsíti.
- **224.0.1.0 – 238.255.255.255** – Používají se globálně v Internetu, mezi organizacemi.
- **239.0.0.0 – 239.255.255.255** – Nejsou globálně viditelné, používají se uvnitř organizací.

Ze speciálních adres stojí za zmínku uvedení tří multicastových adres, které se přímo využívají v protokolu IGMP a v následujících kapitolách na ně narazíme. Jedná se o **224.0.0.1** – adresa pro všechny zařízení (klientské stanice, směrovače) v lokální podsíti, dále adresa **224.0.0.2** – adresuje všechny směrovače v lokální podsíti a **224.0.0.10** – adresuje všechny směrovače s podporou IGMP v3. Kompletní seznam a rozdělení multicastových adres je zveřejněn organizací IANA [2].

Na linkové vrstvě dochází k speciálnímu mapování multicastových IP adres na fyzické adresy. Prvních 25 bitů MAC adresy je pro multicast pevně stanoveno – 01:00:5E. Další 23 bitů se získá z posledních 23 bitů IP adresy (prvních 9 bitů IP adresy je zahazeno). Toto mapování, kdy 9 bitů IP adresy nepoužijeme vytváří situaci, že k jedné MAC adrese se může namapovat 32 multicastových adres.

2.3 Popis multicastové komunikace

V této podkapitole probereme posloupnost akcí, které se na síti odehrávají při multicastové komunikaci. Modelová síť se skládá ze zdroje multicastových dat, klientských stanic jakožto příjemců dat a pak sítí směrovačů, které tvoří cestu mezi zdrojem a příjemci.

Zdroj dat vysílá svá data zabalená do multicastových paketů, které mají zdrojovou adresu běžnou unicastovou adresu a cílovou adresu nastavenou na zvolenou multicastovou adresu značící skupinu. Klienti, kteří se rozhodnou data z této multicastové skupiny přijímat, se pomocí protokolu IGMP přihlásí svému lokálnímu multicastovému směrovači, že mají zájem o data z určené skupiny. Na základě jejich žádosti pak multicastový směrovací protokol vytváří tzv. distribuční strom pro optimální cestu multicastových dat od zdroje k odběrateům a řídí celý přenos. Mezi multicastové směrovací protokoly patří např. PIM, MOSPF, DVMRP.

Směrovače vyhodnocují přeposílání multicastových paketů na další rozhraní nejen podle cílové multicastové adresy, ale i podle zdrojové adresy a to z důvodu zabránění vzniku smyček. Metoda RPF (*Reverse Path Forwarding*) zajišťuje, že směrovač přijímá multicastový provoz jen z těch směrů, které vedou od zdroje vysílání. Při přeposílání paketu na několik rozhraní musí směrovač paket správně duplikovat.

2.4 Multicastové směrování

Pro optimální nalezení cesty od zdroje dat k odběratelům se vytvářejí směrovací protokoly již zmíněné multicastové distribuční stromy.

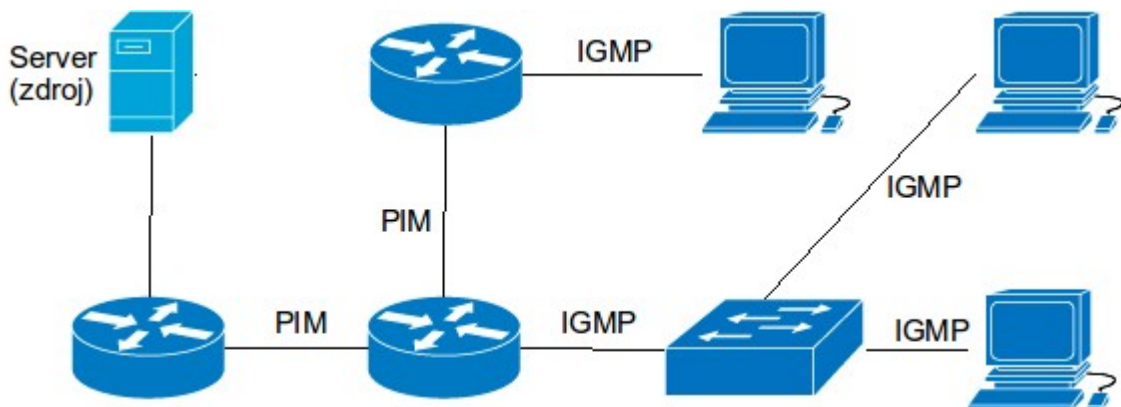
Zdrojový strom má svůj kořen ve zdroji dat a jeho listy tvoří odběratelé. Zdrojový kořen se vždy váže ke zdroji dat, tzn. pro každý zdroj dat musí být vytvořen nový strom. *Sdílený strom* má svůj kořen na určitém místě zvaném *rendesvous point*, ten je pro několik různých zdrojů společný. Listy tvoří opět příjemci multicastových dat. Při změně odběratelů provozu se musí strom přizpůsobit.

Multicastové směrovací protokoly můžeme rozdělit na několik typů:

Dense mode předpokládá, že o multicastová data mají zájem všichni. Směrovače, na jejichž podsítích nejsou odběratelé se zájmem o multicastový provoz, musí svůj nezájem ohlásit explicitně a jejich větve je ořezána. Zástupci *dense mode* protokolů jsou PIM-DM (*Protocol Independent Multicast – Dense Mode*) a DVMRP (*Distance Vector Multicast Routing Protocol*).

Sparse mode je založen na posílání multicastových dat jen těm, kdo si o ně požádají. Příkladem je PIM – SM (*Protocol Independent Multicast – Sparse mode*).

Umístění jednotlivých protokolů v rámci multicastové topologie je na obrázku (Obrázek 2.1).



Obrázek 2.1: Protokoly používané v multicastové síti

3 IGMP

Tato kapitola se zabývá protokolem IGMP – jeho vlastnostmi, funkcionalitou a dále rozdíly mezi jednotlivými verzemi. Největší pozornost je věnována verzi IGMP v2, která je také předmětem implementace rozšiřujícího modulu OMNeT++. V závěru kapitoly jsou uvedeny vlastnosti IGMP na Cisco zařízeních. Hlavním zdrojem informací v této kapitole jsou příslušná RFC jednotlivých verzí IGMP [3][4][5].

3.1 Základní pohled na IGMP

IGMP (*Internet Group Management Protocol*) je komunikační protokol, který slouží ke správě členství klientských stanic v multicastových skupinách. Je prostředníkem mezi klientskou stanicí a lokálním multicastovým směrovačem. Klientská stanice pomocí IGMP informuje nejbližší směrovač v rámci podsítě o svém zájmu přijímat nebo zrušit přijímání multicastových dat z požadovaných multicastových skupin. Směrovač pak na základě klientských požadavků na členství v multicastových skupinách adekvátně zareaguje, předá tuto informaci multicastovému směrovacímu protokolu. Pomocí IGMP zjišťuje zájem klientských stanic o multicastová data. Pro správnou funkčnost musí být tedy IGMP implementován jak na klientské stanici, tak na směrovači.

Nejstarší verze protokolu IGMP v0 vznikla v roce 1986 [6] a už se nepoužívá. V současnosti se lze setkat se 3 verzemi IGMP protokolu – IGMP v1 [3], IGMP v2 [4] a IGMP v3 [5]. Novější verze IGMP jsou zpětně kompatibilní se staršími. Dnes nejužívanějšími jsou IGMP v2 a IGMP v3.

IGMP pracuje na síťové vrstvě ISO/OSI modelu, je součástí protokolu IPv4 - IGMP zprávy jsou zapouzdřeny uvnitř IPv4 datagramu. Hodnota *Protocol* v hlavičce IP datagramu je pro IGMP 2. IGMP zprávy jsou platné jen na lokální podsíti, proto se hodnota *TTL* nastavuje na 1.

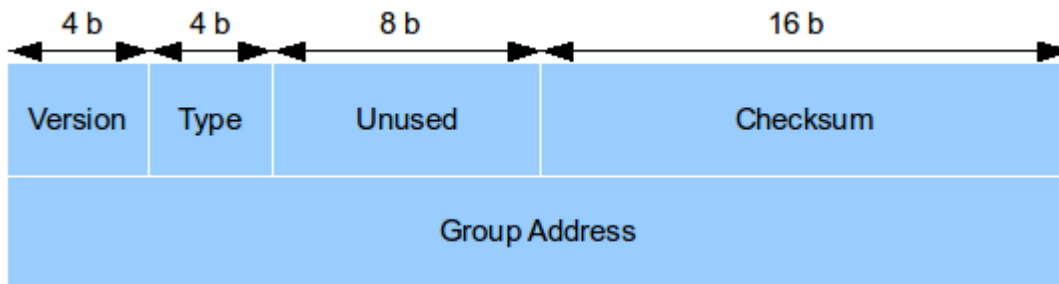
V sítích IPv6 se IGMP nepoužívá, jeho funkcionalitu zastává protokol MLD (*Multicast Listener Discovery*). IGMP v2 odpovídá MLD v1 a IGMP v3 pracuje stejně jako MLD v2.

IGMP zprávy mohou být také zpracovávány přepínači, pokud používají technologii zvanou *IGMP snooping*. Standardně je na přepínači s multicastovými pakety zacházeno jako s broadcastem – přepínač je rozešle na všechny své porty a na větších sítích tak dochází k zbytečnému zahlcování. IGMP snooping si podle průchozích IGMP zpráv sestavuje tabulku s porty a jejich příslušností k multicastovým skupinám. Multicastový provoz pak zasílá je na správné porty, tím omezuje nežádoucí nadbytečný provoz.

3.2 IGMP verze 1

IGMP v1 vznikl v roce 1989 a je definován standardem RFC-1112 [3].

3.2.1 Formát a typy zpráv



Obrázek 3.1: Formát IGMP v1 paketu

Version – verze protokolu (0x1)

Type – typ IGMP zprávy

- *Membership Query* (0x1)
- *Membership Report* (0x2)

Unused – v této verzi protokolu nepoužito

Checksum – kontrolní součet

Group Address – adresa multicastové skupiny

3.2.2 Popis chování

Přihlášení klienta do skupiny – Klientská stanice se přihlašuje do skupiny odesláním zprávy *Membership Report* s adresou požadované multicastové skupiny. Nejbližší směrovač zprávu přijme a k rozhraní, ze kterého *Report* přišel zařadí do požadované multicastové skupiny.

Ověřování zájmu o členství ve skupině – Dotazovací směrovač pravidelně (defaultně po 60 s) pomocí zprávy *Membership Query* na adresu 224.0.0.1 ověřuje zájem klientů o členství v multicastových skupinách. Klienti musí při obdržení zprávy *Membership Query* odpovědět směrovači odesláním zprávy *Membership Report*. Pro omezení počtu zpráv *Membership Report* byl vytvořen následující mechanismus. Klient po příjmu zprávy *Membership Query* vygeneruje náhodně číslo z intervalu 0 - 10. Toto číslo určuje počet sekund, po které klient čeká než odpoví zprávou *Membership Report*. Po tuto dobu také naslouchá, zda jiný klient v síti, který patří do stejné multicastové skupiny, již neposlal svou zprávu *Membership Report*, v tomto případě už klient vlastní

Report neposílá. Pro směrovač stačí jen jeden *Membership Report* za všechny klienty ve stejné multicastové skupině.

Odhlášení klienta ze skupiny – Pro odhlašování neexistuje zvláštní zpráva. Pokud už nechce klient přijímat data z multicastové skupiny, přestane odpovídat na ověřovací zprávy *Membership Query*. Směrovač posílá zprávy *Membership Query* maximálně 3x za sebou. Pokud se žádný klient do této doby směrovači neozve, směrovač přestane posílat multicastová data do této části sítě. Tento způsob odhlašování však trvá delší dobu (3x *Query Interval*) a během této doby se zbytečně posílají do dané části sítě multicastová data.

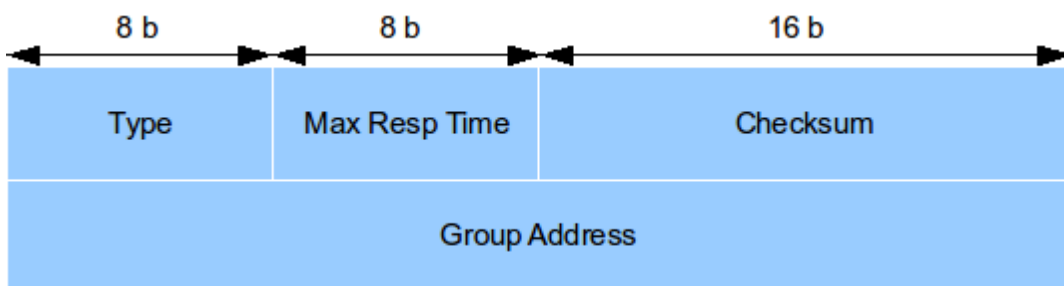
Volba dotazovacího směrovače – Pokud se na dané podsíti vyskytne více směrovačů, jen jeden se může stát tzv. dotazovačem (*Querier*). Dotazovací směrovač je v této verzi protokolu vybrán multicastovým směrovacím protokolem.

3.3 IGMP verze 2

IGMP v2 byl vytvořen standardem RFC-2236 [4] v roce 1997. Oproti první verzi obsahuje několik změn a vylepšení.

Došlo ke drobné změně ve formátu zprávy – pole *Version* a *Type* z první verze byly sloučeny do pole *Type*. Namísto nevyužitého pole přibylo *Max Resp Time*. Protokol nově řídí volbu *Querier* směrovače. Byly také vytvořeny nové typy zpráv. Pro odhlašování ze skupiny byla vytvořena zpráva *Leave Group*, která zkracuje proces odhlašování. Pro oznamování klientem byla vytvořena *Membership Report v2*. *Membership Query* se rozdělilo na *General Query* a *Group-Specific Query*. Hlavička IP navíc u zpráv IGMP v2 obsahuje *IP Router Alert Option*, který označuje, že směrovač musí paket zpracovat.

3.3.1 Formát a typy zpráv



Obrázek 3.2: Formát IGMP v2 paketu

Type – typ IGMP zprávy – pro zpětnou kompatibilitu zpráv z IGMP v1 slučuje hodnoty *Version* a *Type* z IGMP v1.

- *Membership Query* (0x11)
- *Membership Report v1* (0x12) – zpětná kompatibilita s IGMP v1
- *Membership Report v2* (0x16)
- *Leave Group* (0x17)

Max Resp Time – používá se u zpráv *Membership Query*, označuje maximální čas v desetinách sekund, po které klient čeká než odešle odpověď směrovači zprávou *Membership Report*. U ostatních zpráv je nastaven na 0 a jeho hodnota se ignoruje.

Checksum – kontrolní součet

Group Address – adresa multicastové skupiny

Následující výčet popisuje všechny typy zasílaných zpráv, jejich užití a v popisných tabulkách je znázorněn obsah důležitých polí v IGMP zprávách a IP hlavičce.

Membership General Query – tuto zprávu posílá směrovač klientským stanicím, aby zjistil jejich zájem o odběr dat z multicastových skupin.

Membership General Query	
Type	0x11
Max Resp Time	<i>Query Response Interval</i> (defaultně 100 ... 10 s)
Group Address	0 (ignoruje se)
Zrojová IP	IP směrovače
Cílová IP	224.0.0.1 (all systems)

Tabulka 3.1: Zpráva Membership General Query

Membership Group-Specific Query – tuto zprávu posílá směrovač klientským stanicím po obdržení zprávy *Leave Group*, aby zjistil, zda se na síti nenachází ještě nějaký klient se zájmem o danou multicastovou skupinu.

Membership Group-Specific Query	
Type	0x11
Max Resp Time	<i>Last Member Query Interval</i> (defaultně 10 ... 1 s)
Group Address	Adresa multicastové skupiny
Zrojová IP	IP směrovače
Cílová IP	Adresa multicastové skupiny

Tabulka 3.2: Zpráva Membership Group-Specific Query

Membership Report – tuto zprávu posílá klientská stanice při zájmu o odběr dat určité multicastové skupiny. Verze 1 a 2 se liší jen číslem typu.

Membership Report v1	
Type	0x12
Max Resp Time	0 (ignoruje se)
Group Address	Adresa multicastové skupiny
Zrojová IP	IP klienta
Cílová IP	Adresa multicastové skupiny

Tabulka 3.3: Zpráva Membership Report v1

Membership Report v2	
Type	0x16
Max Resp Time	0 (ignoruje se)
Group Address	Adresa multicastové skupiny
Zrojová IP	IP klienta
Cílová IP	Adresa multicastové skupiny

Tabulka 3.4: Zpráva Membership Report v2

Leave Group – tuto zprávu posílá klientská stanice při ukončení odběru o data z dané multicastové skupiny.

Leave Group	
Type	0x17
Max Resp Time	0 (ignoruje se)
Group Address	Adresa multicastové skupiny
Zrojová IP	IP klienta
Cílová IP	224.0.0.2 (all routers)

Tabulka 3.5: Zpráva Leave Group

3.3.2 Proměnné a časovače

IGMP v2 definuje řadu proměnných a časovačů, které jsou pak využity při implementaci tohoto protokolu.

- **Robustness Variable** – Základní proměnná, která se využívá k výpočtu několika dalších časovačů a hodnot. Nesmí být 0 a neměla by být 1. Výchozí hodnota je 2.

- **Query Interval** – Jedná se o interval mezi posíláním zpráv *General Query*. Výchozí hodnota je 125 sekund.
- **Query Response Interval** – Tato hodnota se vkládá do pole *Max Resp Time* ve zprávách *Membership General Query*. Značí maximální čas zpoždění odpovědi klienta. Hodnota musí být menší než *Query Interval*. Výchozí hodnota je 100, což odpovídá 10 sekundám.
- **Group Membership Interval** – Jedná se dobu, po kterou si směrovač pamatuje členství skupiny, pak členství zruší.
Hodnota: $(Robustness\ Variable \cdot Query\ Interval) + Query\ Response\ Interval$
- **Other Querier Present Interval** – Označuje dobu, která musí uplynout než směrovač na určitém rozhraní přejde ze stavu *Non-Querier* do stavu *Querier*.
Hodnota: $(Robustness\ Variable \cdot Query\ Interval) + (Query\ Response\ Interval \div 2)$
- **Startup Query Interval** – Jedná se o interval mezi zprávami *Membership General Query*, které posílá směrovač při inicializaci.
Výchozí hodnota: $Query\ Interval \div 4$
- **Startup Query Count** – Hodnota označuje počet inicializačních zpráv *Membership General Query*. Výchozí hodnota je *Robustness Variable*.
- **Last Member Query Interval** – Tato hodnota se vkládá do pole *Max Resp Time* ve zprávách *Membership Group-Specific Query*, které se posílají jako odpověď na *Leave Group*. Také tato hodnota označuje interval mezi zprávami *Membership Group-Specific Query*. Výchozí hodnota je 10, což odpovídá 1 sekundě.
- **Unsolicited Report Interval** – Označuje interval odesílání inicializačních zpráv *Membership Report* při prvotním přihlášení klienta do multicastové skupiny. Výchozí hodnota je 10 sekund.
- **Version 1 Router Present Timeout** – Při zajištění kompatibility mezi IGMP v1 směrovačem a IGMP v2 klientem. Pokud klient od směrovače obdrží *Membership Query v1* musí odesílat *Membership Report v1* a až po uplynutí časovače *Version 1 Router Present Timeout* může poslat zprávu *Membership Report v2*. Hodnota je 400 sekund.

3.3.3 Popis chování

Směrovače musí mít pro každé fyzické rozhraní, na kterém je dostupný protokol IGMP, uloženo několik údajů – seznam připojených multicastových skupin, přidružené časovače a role rozhraní (*Querier* nebo *Non-Querier*). Klientské stanice musí mít také pro každé fyzické rozhraní uloženy seznam multicastových skupin s časovači a příznak posledního odběratele skupiny.

Inicializace směrovače – Po spuštění IGMP na určitém rozhraní směrovače vyšle směrovač inicializační zprávu *Membership General Query* určenou všem klientům připojeným na dané

rozhraní s cílem zjistit jejich zájem o odběr multicastových dat. Zprávu znovu vyšle v intervalu *Startup Query Interval*, celkový počet inicializačních *General Query* zpráv se řídí hodnotou *Startup Query Count*. Poté se směrovač dostane do běžného režimu.

Volba dotazovacího směrovače – Každé fyzické rozhraní směrovače může být *Querierem*, který pravidelně vysílá zprávy *Membership General Query* nebo *Non-Querierem*, který tyto zjišťovací zprávy nezasílá a jeho úlohu plní na síti jiný směrovač. *Non-Querier* také ignoruje zprávy typu *Leave Group*.

Ve výchozím stavu se každé rozhraní směrovače nachází ve stavu *Querier*. Pokud však směrovač obdrží na nějakém rozhraní zprávu *Membership Query* od směrovače s nižší IP adresou, přestává být *Querierem* a stává se *Non-Querierem*. Pokud v intervalu daným hodnotou *Other Querier Present Interval* neobdrží směrovač na daném rozhraní žádné další zprávy *Membership Query* od směrovače s nižší IP adresou, přechází zpět do výchozího stavu *Querier*.

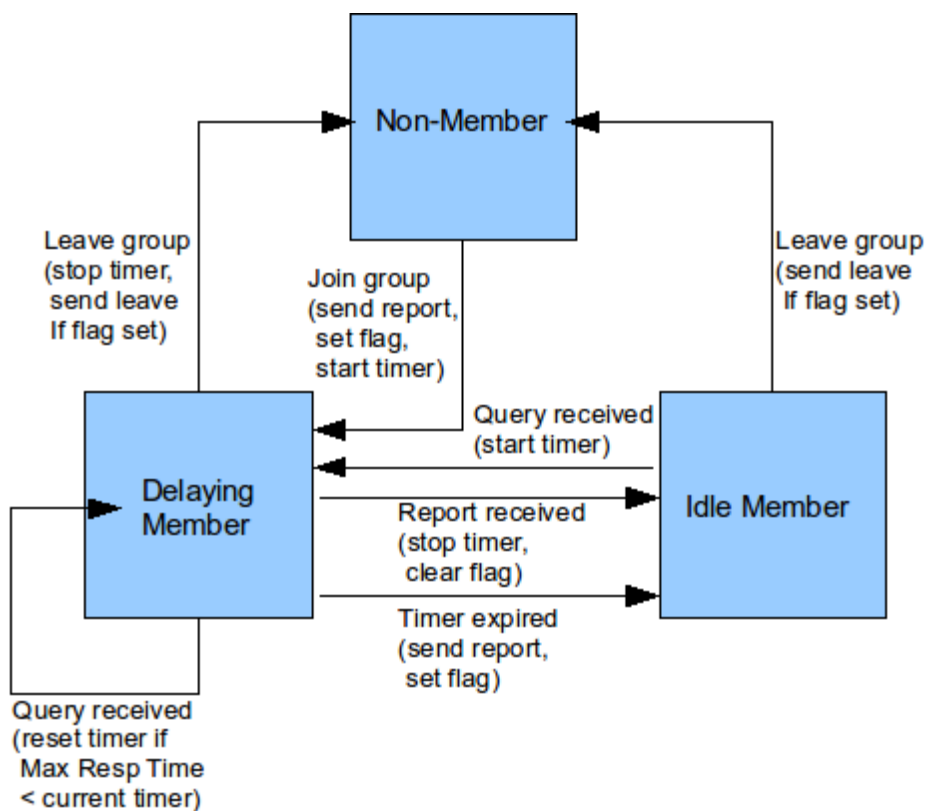
Přihlášení klienta do skupiny – Pokud se chce klient přihlásit k nějaké multicastové skupině, odešle nevyžádanou zprávu *Membership Report v2*. Aby se předešlo možné ztrátě zprávy, klient zprávu znovu pošle po uplynutí intervalu *Unsolicited Report Interval*. Směrovač zprávu *Membership Report v2* přijme a zkontroluje, zda už tato skupina není k rozhraní přiřazena, pokud ne, tak si skupinu zařadí do tabulky členství. Zároveň spustí k této skupině časovač členství s hodnotou *Group Membership Interval*. Pokud během doby běhu časovače nepřijde na rozhraní další zpráva *Membership Report* s požadavkem na stejnou multicastovou skupinu, směrovač vyřadí záznam z tabulky členství a ukončí posílání dat na rozhraní. Pokud přijde zpráva *Membership Report* ještě před vypršením časovače, hodnota časovače se resetuje na výchozí hodnotu *Group Membership Interval*.

Ověřování zájmu o členství ve skupině – Směrovač pravidelně v intervalu *Query Interval* posílá zprávy *Membership General Query*. Pokud klient tuto zprávu přijme, nastaví si pro každou odebíranou skupinu časovač s náhodnou hodnotou v rozsahu 0 až *Max Response Time* získaný z příchozí *Membership Query*. Pokud je už časovač nějaké skupiny aktivní a hodnota *Max Resp Time* je menší než zbývající doba časovače, je časovač restartován, jinak se nechá doběhnout do konce. Po vypršení tohoto časovače zpoždění vyšle klient na danou skupinu *Membership Report v2*. Aby se předcházelo posílání zbytečných reportů, pokud klient přijme *Membership Report* od jiného klienta na stejnou multicastovou skupinu, tak klient časovač příslušný k dané skupině zastaví a svůj *Membership Report* už neposílá.

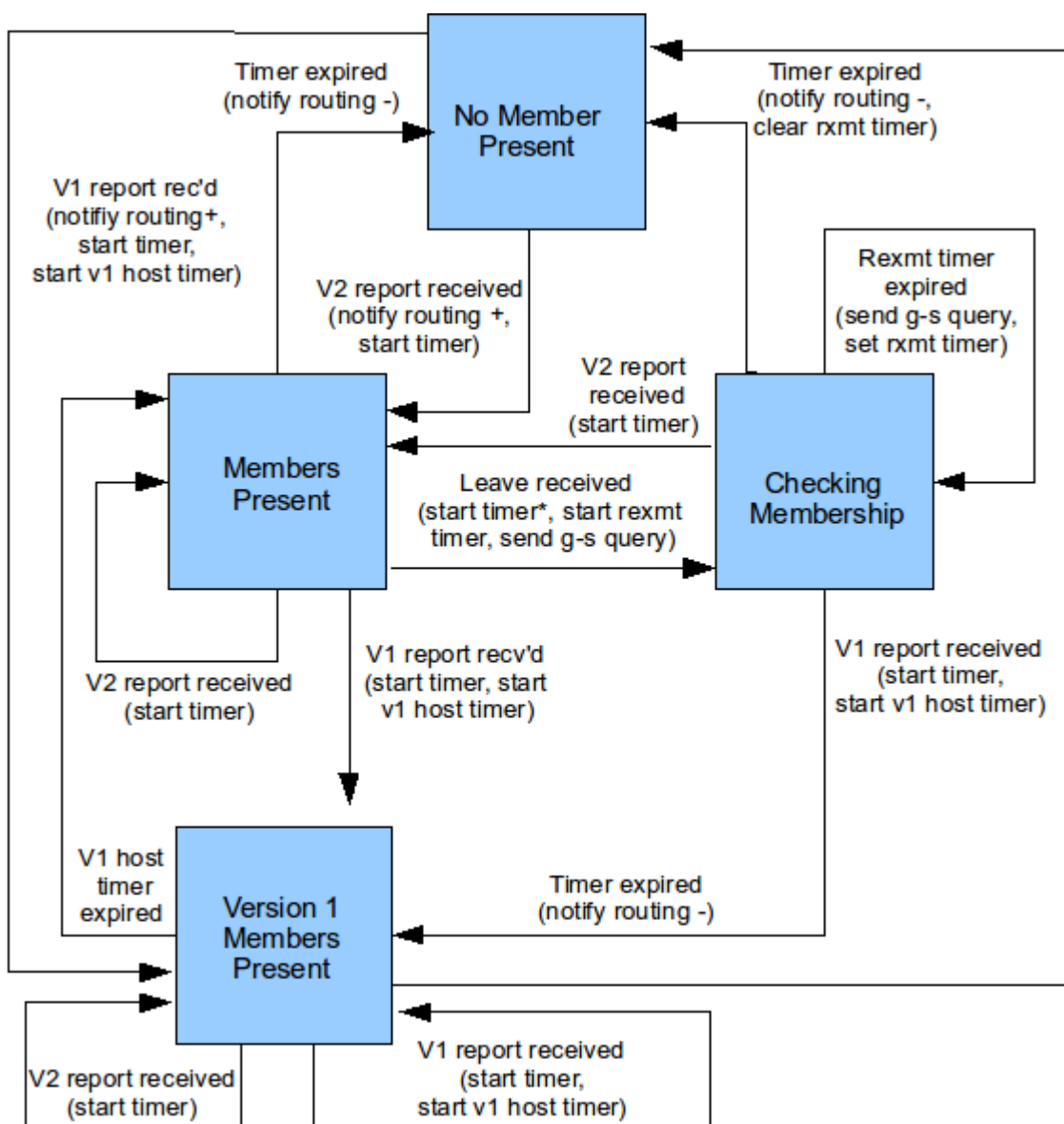
Odhlášení klienta ze skupiny – V případě, že chce klient zrušit odebrání dat z multicastové skupiny, odešle směrovači zprávu *Leave Group*. Směrovač ještě před odstraněním záznamu o členství multicastové skupiny na daném rozhraní odešle zprávu *Membership Group-Specific Query* s cílem zjistit, zda se ještě na daném segmentu sítě nenachází ještě nějaký klient se zájmem o danou skupinu. Tuto zprávu odešle celkem v počtu *Last Member Query Count* v intervalu *Last Member Query Interval*. Pokud se během doby nastavené v *Max Resp Time* zprávy neozve žádný jiný klient, který by chtěl na daném rozhraní stejnou multicastovou skupinu odebrat, směrovač skupinu vyřadí ze své tabulky.

Odhlásování klienta z odběru dat určité multicastové skupiny závisí na implementaci IGMP. Klient může být implementován tak, že naslouchá zprávám *Membership Report* od ostatních klientů a pamatuje si, zda ještě nějaký klient odpovídal na zprávu *Membership Query* se stejnou multicastovou skupinou. Pokud tedy klient ví, že se na síti nachází ještě jiný klient, který je přihlášený ke stejné multicastové skupině, nemusí se už odhlášovat zprávou *Leave Group*. Naopak, pokud klient ví, že je poslední na síti, který odebrá určitou multicastovou skupinu, pak je odhlášení zprávou *Leave Group* nutné. Pokud je implementace IGMP na klientovi taková, že si informace o ostatních klientech neuchovává, musí odesílat zprávu *Leave Group* vždy.

Na obrázcích (3.3, 3.4) jsou uvedeny stavové diagramy pro klientské stanice a směrovače.



Obrázek 3.3: Stavový diagram IGMP pro klienty (převzato z [4])



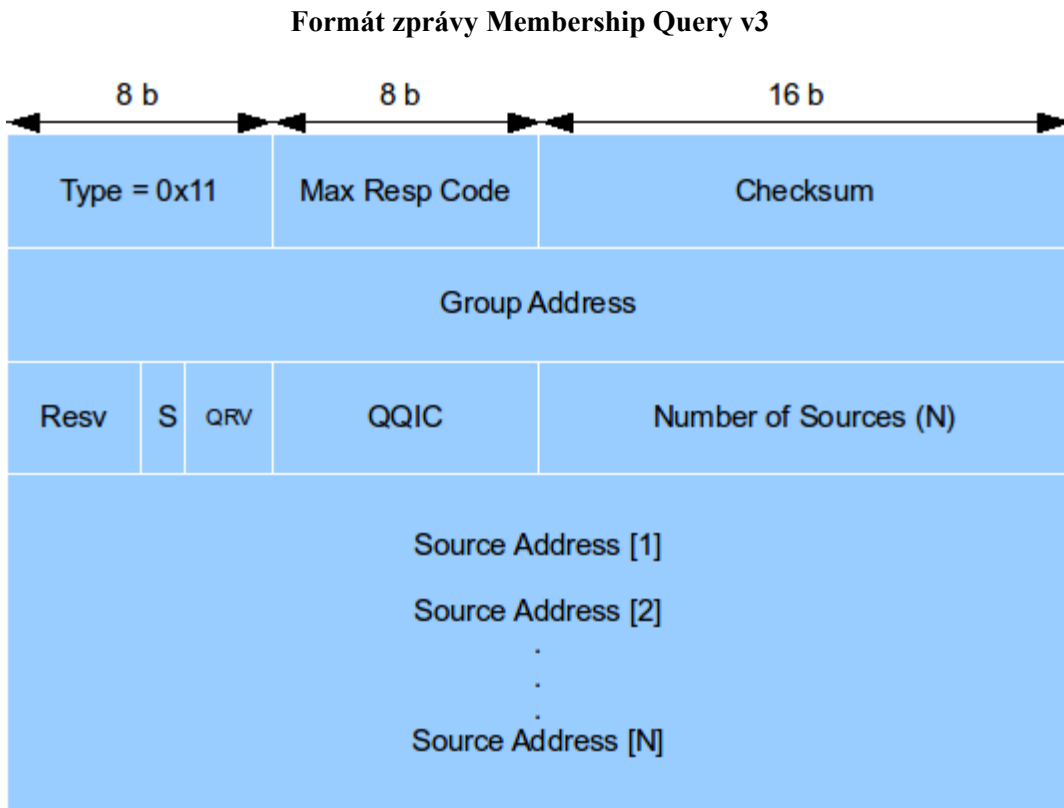
Obrázek 3.4: Stavový diagram IGMP směrovače (převzato z [4])

3.4 IGMP verze 3

IGMP v3 byl definován standardem RFC-3376 [5] v roce 2002. Je nejnovější verzí protokolu IGMP. Oproti předchozím verzím již nemají zprávy fixní délku. Inovací od předchozích verzí je možnost specifikace zdrojů multicastového vysílání (používá se při *Source Specific Multicast*). I třetí verze protokolu si zachovává kompatibilitu s předchozími verzemi.

3.4.1 Formát a typy zpráv

IGMP v3 používá jen dvě vlastní zprávy – *Membership Query v3* (Obrázek 3.5) a *Membership Report v3* (Obrázek 3.6 a Obrázek 3.7).



Obrázek 3.5: Formát zprávy Membership Query v3

Type – *Membership Query* (0x11) , máme 3 druhy zpráv:

- *General Query* – pole *Group Address* a *Number of Sources* je 0. Platí pro všechny stanice na podsíti, nezáleží ke které multicastové skupině patří. Posílá se na adresu 224.0.0.1.
- *Group-specific Query* – pole *Group Address* obsahuje multicastovou adresu skupiny, pole *Number of Sources* je 0; platí pro všechny stanice určité multicastové skupiny na podsíti.
- *Group-and-source-specific Query* – dotaz se specifikací skupiny a zdroje.

Max Resp Code – označuje maximální počet desetín sekund, po které klient čeká než odešle odpověď routeru zprávou *Membership Report*. Pokud je hodnota *Max Resp Code* nižší než 128, použije se uvedená hodnota, pokud je vyšší nebo rovna 128, hodnota se interpretuje jako exponent (1. - 3. bit) a mantisa (4. - 7. bit).

Checksum – kontrolní součet

Group Address – při *Membership General Query* 0.0.0.0, při *Membership Group-specific Query* nebo *Membership Group-and-Source-Specific Query* obsahuje multicastovou adresu skupiny.

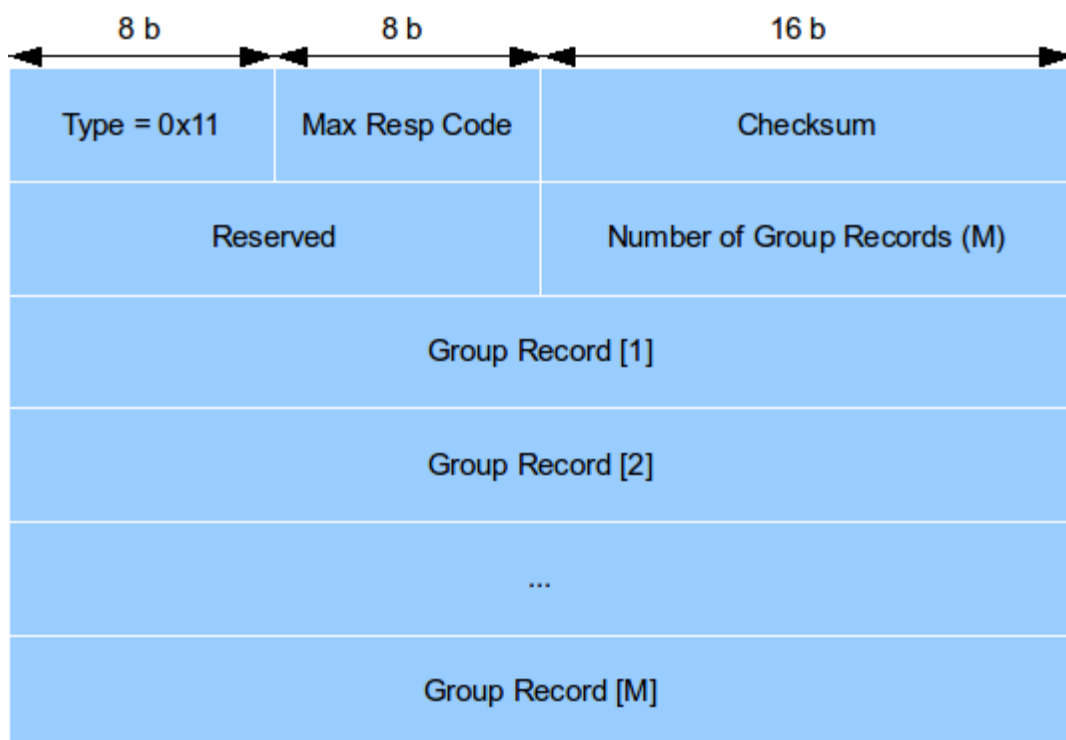
Resv - rezervováno

QRV (Querier's Robustness Variable) – obsahuje proměnnou *Robustness Variable* používanou *Querier* směrovačem. Směrovače tuto hodnotu převezmou namísto výchozí hodnoty. Pokud je hodnota 0, používá se výchozí hodnota.

QQIC (Querier's Query Interval Code) – obsahuje proměnnou *Query Interval* používanou *Querier* směrovačem. *Non-Querier* směrovače tuto hodnotu převezmou namísto výchozí hodnoty. Pokud je hodnota 0, používá se výchozí hodnota. V případě hodnoty rovné nebo vyšší než 128 je interpretace jako exponent a mantisa jako u *Max Resp Code*.

Number of Sources (N) – obsahuje počet zdrojových adres jen u zprávy *Membership Group-and-Source-Specific Query*. U ostatních zpráv je hodnota 0.

Formát zprávy Membership Report v3



Obrázek 3.6: Formát zprávy Membership Report v3

Source Address [N] – vektor zdrojových unicastových adres, počet je dán předchozím polem.

Type – typ zprávy:

- *Membership Report v1* (0x12) – zpětná kompatibilita s IGMP v1
- *Membership Report v2* (0x16) – zpětná kompatibilita s IGMP v2

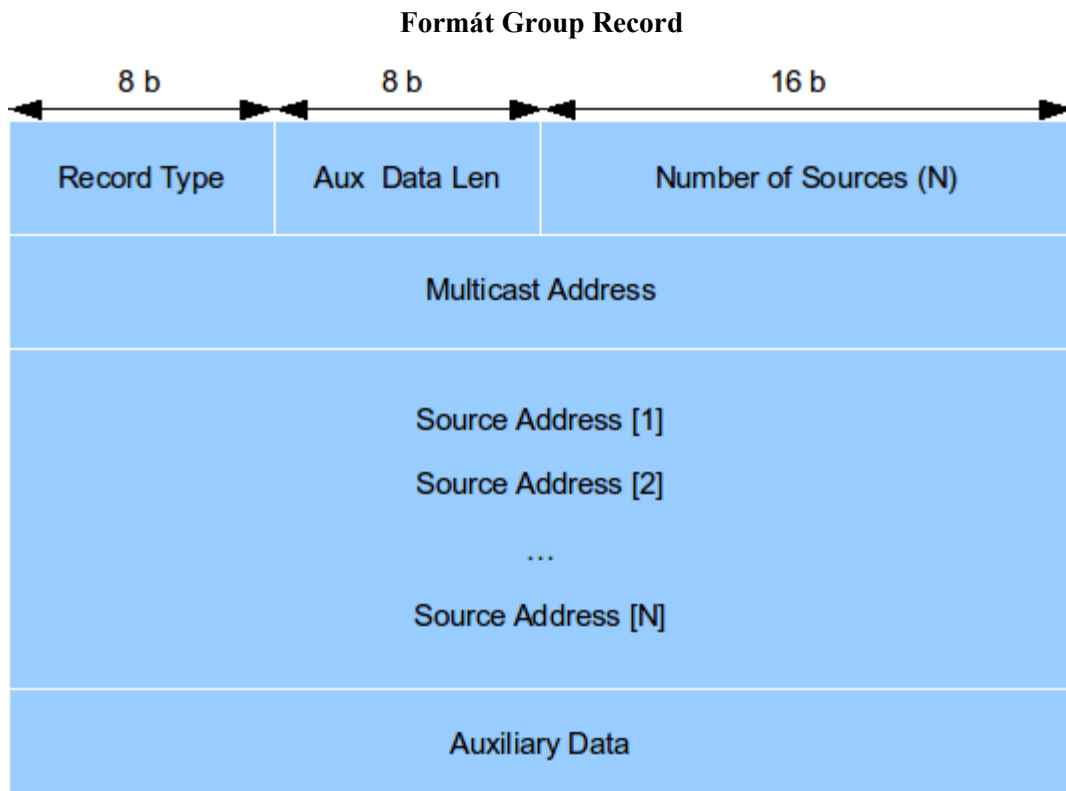
- *Membership Report v3* (0x22) – posílá se na adresu 224.0.0.22
- *Leave Group* (0x17) – zpětná kompatibilita s IGMP v2

Reserved – rezervováno

Checksum – kontrolní součet

Number of Groups Record (M) – Počet záznamů o skupině.

Groups Record – obsahuje záznamy o skupině, podrobný popis níže.



Obrázek 3.7: Formát Group Record

Record Type – existuje několik typů:

- *Current-State Record* – klient tímto odpovídá na *Membership Query v3*. Vrací aktuální filtrovací mód a seznam zdrojových adres:
 - *MODE_IS_INCLUDE* (1) – typ filtrování zdroje je INCLUDE.
 - *MODE_IS_EXCLUDE* (2) – typ filtrování zdroje je EXCLUDE.
- *Filter-Mode-Change Record* – klient posílá při změně filtrovacího módu:
 - *CHANGE_TO_INCLUDE_MODE* (3) – změna na mód INCLUDE, v *Source Address* je nový seznam adres zdrojů.

- *CHANGE_TO_EXCLUDE_MODE* (4) - změna na mód EXCLUDE, v *Source Address* je nový seznam adres zdrojů.
- *Source-List-Change Record* – klient posílá při změně v *Source Address*:
 - *ALLOW_NEW_SOURCES* (5) – v *Source Address* budou nové adresy zdrojů, od kterých chceme přijímat provoz. V případě módu INCLUDE to budou adresy zdrojů, které přidáme na seznam zdrojů. V případě módu EXCLUDE to budou adresy zdrojů, které ze seznamu zdrojů odebereme.
 - *BLOCK_OLD_SOURCES* (6) – v *Source Address* budou adresy zdrojů, od kterých nechceme přijímat provoz. V případě módu INCLUDE to budou adresy zdrojů, které odebereme ze seznamu zdrojů. V případě módu EXCLUDE to budou adresy zdrojů, které chceme přidáme do seznamu zdrojů.

Aux Data Len – délka pomocných dat v poli *Auxiliary Data*. Pokud žádná další neobsahuje, je hodnota 0.

Number of Groups Record (M) – počet záznamů o skupině

Multicast Address – adresa skupiny

Source Address - vektor IP adres zdrojů multicastového provozu

Auxiliary Data – pro budoucí využití, zatím se ignoruje

3.4.2 Popis chování

Filtrování zdrojů – Novou funkcionalitou je tzv. filtrování zdrojů. Používá se při situaci, kdy se na síti nachází více zdrojů multicastového provozu pro stejnou multicastovou skupinu. Filtrování může probíhat dvěma způsoby. Mód INCLUDE zajišťuje příjem multicastového provozu pouze z vyjmenovaných zdrojů z dané multicastové skupiny. Naopak mód EXCLUDE zajišťuje příjem multicastových dat ze všech zdrojů z dané multicastové skupiny kromě vyjmenovaných.

Přihlašování do skupiny – Přihlašování do multicastové skupiny probíhá odesláním *Membership Report v3* na adresu 224.0.0.22, kde sídlí všechny IGMP v3 směrovače. Pro přihlášení se *Group Record* nastavuje na mód EXCLUDE s prázdnou množinou zdrojových adres.

Změna stavu rozhraní – Klient může podle potřeb měnit zdrojové adresy multicastového provozu v dané skupině, může měnit mód INCLUDE/EXCLUDE. Posílá zprávy *Membership Report v3* s příslušným nastavením *Group Record*.

Odhlašování ze skupiny – Odhlašování probíhá opět pomocí zpráv *Membership Report v3* tentokrát změnou stavu skupiny na INCLUDE s prázdnou množinou zdrojových adres.

3.5 IGMP na Cisco zařízeních

Tato kapitola obsahuje stručný přehled možností a konfigurace IGMP [7] na síťových zařízeních firmy Cisco.

Pro konfiguraci multicastu, potažmo IGMP, nastavíme následující příkazy. Nejdříve musíme zapnout multicastové směrování:

```
Router(config)# ip multicast-routing
```

V dalším kroku zapneme směrování PIM, kterým zároveň aktivujeme IGMP.

```
Router(config-if)# ip pim {dense-mode | sparse-dense-mode | sparse-mode}
```

Dalšími příkazy pro konfiguraci PIMu se zde zabývat nebudeme. Lze je například vyčíst z [7].

Po aktivaci PIMu je na daném rozhraní spuštěna druhá verze IGMP. Pro změnu verze se používá příkaz `ip igmp version`.

```
Router(config-if)# ip igmp version {1 | 2 | 3}
```

Příkaz `ip igmp join-group` umožňuje explicitně připojit multicastovou skupinu na dané rozhraní.

```
Router(config-if)# ip igmp join-group 226.100.100.5
```

Příkaz `ip igmp static-group` přiřadí multicastovou skupinu na dané rozhraní staticky. Tím odpadá zpracovávání přihlašování a odhlašování klientů, multicastový provoz se bude šířit na toto rozhraní vždy. V určitých aplikacích přinášení statické nastavení skupin zrychlení provozu.

```
Router(config-if)# ip igmp static-group 226.100.100.5
```

Dále můžeme konfigurovat některé časovače.

```
Router(config-if)# ip igmp query-interval  
Router(config-if)# ip igmp querier-timeout  
Router(config-if)# ip igmp query-max-response-time  
Router(config-if)# ip igmp last-member-query-interval  
Router(config-if)# ip igmp last-member-query-count
```

Na IGMP můžeme aplikovat ACL.

```
Router(config-if)# ip igmp access-group
```

Můžeme také nastavit směrovač, na který se pak klientské zprávy *Membership Report* a *Leave Group* budou přeposílat.

```
Router(config-if)# ip igmp helper-address
```

Příkaz `show ip igmp group` nám umožňuje vypsat seznam multicastových skupin na daném rozhraní, můžeme zjistit aktuální hodnotu časovače *Group Membership Interval* a IP adresu klienta, který se ohlásil naposledy.

```
Router# show ip igmp group
IGMP Connected Group Membership
Group Address  Interface          Uptime  Expires  Last Reporter
226.100.100.5  FastEthernet0/1   0d0h20m 00:01:32 192.168.1.8
```

Kompletní informace o IGMP na daném rozhraní lze získat příkazem `show ip igmp interface`. Najdeme zde základní informace o rozhraní, verzi protokolu IGMP, hodnoty časovačů, adresu *Querier* směrovače a připojené multicastové skupiny.

```
Router>show ip igmp interface fastethernet0/1
FastEthernet0/1 is up, line protocol is up
Internet address is 192.168.0.5, subnet mask is 255.255.255.0
IGMP is enabled on interface
Current IGMP version is 2
CGMP is disabled on interface
IGMP query interval is 60 seconds
IGMP querier timeout is 120 seconds
IGMP max query response time is 10 seconds
Inbound IGMP access group is not set
Multicast routing is enabled on interface
Multicast TTL threshold is 0
Multicast designated router (DR) is 192.168.0.5 (this system)
IGMP querying router is 192.168.0.5 (this system)
Multicast groups joined: 226.100.100.5 227.100.100.1
```

Na Cisco směrovačích je také rozdílná volba *Querier* směrovače než jak je popsána v RFC – *Querierem* se stává směrovač s vyšší IP adresou.

4 OMNeT++

Tato kapitola seznamuje čtenáře se simulačním nástrojem OMNeT++, knihovnou INET a jejími možnostmi při simulaci multicastu. Informace z této kapitoly pocházejí z manuálu [8] OMNeT++.

4.1 OMNeT++

OMNeT++ [9] je objektově orientovaný diskretní simulátor pro simulaci nejen počítačových sítí. Kromě vlastního simulátoru zahrnuje nástroj také integrované vývojové prostředí a uživatelsky přívětivé GUI. Autorem OMNeT++ je András Varga. Nástroj je vytvořený v C++, pracuje pod operačními systémy GNU/Linux, Mac OS/X, Windows. Je volně šiřitelný pro nekomerční použití. Pro komerční účely je nabízen pod názvem OMNEST.

OMNeT++ je modulárním systémem, proto je snadno rozšiřitelný a přizpůsobitelný pro různé druhy simulací. Má hierarchickou strukturu, moduly se vzájemně propojují pomocí tzv. bran, více jednoduchých modulů dohromady může vytvořit tzv. složený modul. Mezi sebou moduly komunikují zasíláním zpráv.

Pro popis modelu se používá vlastní jazyk zvaný NED. V `.ned` souborech tedy definujeme strukturu modelu, složení z jednotlivých modulů, brány a různé parametry. Zprávy pro komunikaci mezi moduly vytváříme pomocí souborů `.msg`. Můžeme si nadefinovat vlastní datové struktury dle potřeby. Vlastní moduly vytváříme pomocí C++ v souborech `.cc` a `.h`.

4.2 INET framework

INET framework [10] je rozšíření simulátoru OMNeT++ o moduly pro simulace IP sítí. Najdeme zde moduly implementující protokoly TCP, UDP, IPv4, Ipv6, MPLS, Ethernet, PPP a jiné. Pro simulaci multicastu však potřebné protokoly implementovány nejsou.

5 Popis implementace

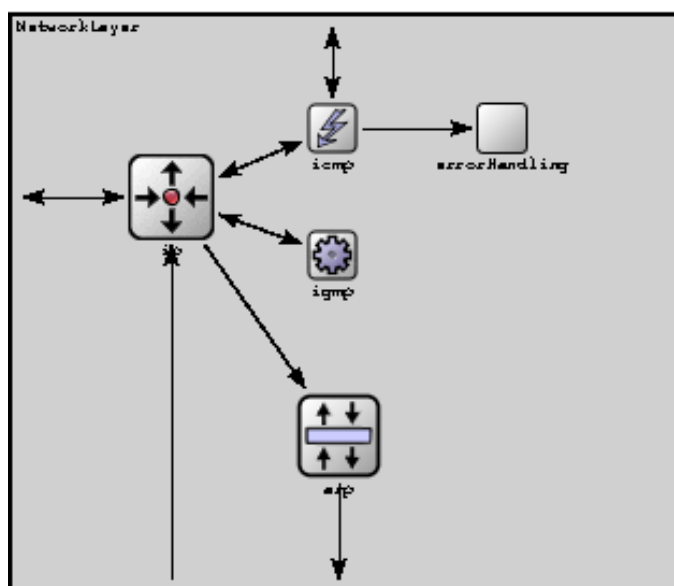
Tato kapitola popisuje návrh a implementaci nového modulu IGMP do frameworku INET.

5.1 Návrh modulu

Základním rozhodnutím při návrhu nového modulu byla volba implementované verze protokolu. Verze IGMP v1 byla pro svou zastaralost vyřazena hned, zvažovalo se mezi druhou a třetí verzí. Protože jsme při implementaci podpory multicastu do OMNeT++ teprve v začátku, byla po úvaze zvolena verze IGMP v2. Až se podaří rozšířit simulátor o protokol PIM a odladit funkční multicastovou komunikaci, můžeme uvažovat o rozšíření do novější verze protokolu.

Protože se zabýváme počítačovým modelováním a nepotřebujeme všechny vlastnosti reálného systému, určili jsme si rozsah implementace našeho modulu. Některé vlastnosti popsané v RFC, z našeho pohledu nevyužitelné, jsme vypustili. Naším cílem bylo vytvořit část protokolu na směrovači, logiku na klientských stanicích neuvažujeme. Také jsme se rozhodli nezahrnout zpětnou kompatibilitu s IGMP v1, protože v našich simulacích nebudeme mít klienty vysílající zprávy z první verze protokolu.

Ze stávající implementace knihovny INET vyplývá, že modul IGMP bude zakomponován na předpřipravené místo – jako submodul modulu NetworkLayer (Obrázek 5.1) a přes bránu připojený k modulu IP. Autory byl navržen IGMP modul jako složený, ve skutečnosti není důvod IGMP členit na několik samostatných podmodulů, proto byl navržen jen jednoduchý modul.



Obrázek 5.1: Umístění modulu IGMP jako součásti modulu NetworkLayer

Modul IGMP bude v budoucnu spolupracovat s modelem PIM. Při návrhu byl na tento fakt brán zřetel a modul bude schopen podle požadavků modulu PIMu poskytnout potřebné údaje (např. informace o právě přihlášené nebo odhlášené skupině). Protože však modul PIMu nebyl vyvíjen současně, bude třeba konkrétní spolupráci ještě doimplementovat. To se také týká konfigurace IGMP na rozhraních, které je také svázáno s PIMem.

5.2 Implementace modulu

Implementace se skládá z těchto souborů:

- IGMP.ned – popis modulu
- IGMP.cc – zdrojový soubor
- IGMP.h – hlavičkový soubor
- IGMPMessage.msg – definice IGMP zpráv
- IGMPTimer.msg – definice časovačů

Protože je IGMP součástí modulu NetworkLayer, který je používán společně v topologiích směrovače a koncových stanic, bylo nutné rozlišit, zda se modul nachází ve směrovači. K tomu byla vytvořena funkce `isRouter()`, která se používá pro označení části kódu, který se provede jenom na směrovači.

5.3 Zprávy

Základním stavebním prvkem IGMP modulu je definování formátu IGMP zprávy `IGMPMessage.msg`. OMNeT++ zprávu přeloží a vygeneruje příslušný zdrojový a hlavičkový soubor. Naplnění zprávy potřebným obsahem je úkolem funkce `createPacket()`. Odesílání jednotlivých zpráv je pak řešeno funkcemi:

- `sendMembershipGeneralQuery()`
- `sendMembershipGroupSpecificQuery()`
- `sendMembershipReportV2()`
- `sendLeaveGroup()`.

Vygenerovaná zpráva je pak odeslána do modulu IP, který ji odešle přes zvolené síťové rozhraní směrovače do sítě. Naopak, při přijetí zprávy ze sítě dorazí do modulu IP, ten ji pak podle čísla protokolu (pro IGMP = 2) přepoše modulu IGMP. V modulu IP (`IP.cc`) byla provedena úprava zdrojového kódu, protože modul při příjmu zpráv nezohledňoval číslo protokolu a zprávy zahazoval.

Obsah zprávy `IGMPMessage.msg`:

```
enum IGMPType
{
    IGMP_MEMBERSHIP_QUERY = 0x11;
    IGMP_MEMBERSHIP_REPORT_V1 = 0x12;
    IGMP_MEMBERSHIP_REPORT_V2 = 0x16;
    IGMP_LEAVE_GROUP = 0x17;
};

packet IGMPMessage
{
    short type enum(IGMPType);
    short maxRespTime;
    IPAddress groupAddress;
};
```

5.4 Časovače

Časovače slouží k odpočítávání různých událostí, jsou implementovány jako zprávy, které posílá modul IGMP sám sobě s požadovaným časovým zpožděním. V reakci na příchozí zprávu je pak vykonána požadovaná akce.

Ukazatelé na právě běžící časovače se ukládají do tabulky IGMP rozhraní, aby byly dobře dostupné v případě potřeby časovač smazat či resetovat. Pro vytvoření či vyresetování časovačů byly vytvořeny tyto funkce:

- `startInterfaceTimer()`
- `startGroupMembershipTimer()`
- `startLastMemberTimer()`

Časovač `IGMPTimer` je definována v `IGMPTimer.msg`.

```
enum IGMPTimerType
{
    TIMER_INIT_GENERAL_QUERY = 1;
    TIMER_GENERAL_QUERY = 2;
    TIMER_OTHER_QUERIER_PRESENT = 3;
    TIMER_GROUP_MEMBERSHIP = 4;
```

```

    TIMER_LAST_MEMBER = 5;
};

message IGMPTimer extends cMessage
{
    char timerKind enum(IGMPTimerType);
    int count;
    int intID;
    IPAddress groupAddress;
};

```

5.5 Rozhraní IGMP

Směrovače si musí pro každé své rozhraní, na kterém běží IGMP, uchovávat několik informací. Standardně je pro uchování základních informací o rozhraní určen modul InterfaceTable se záznamem o rozhraní InterfaceEntry. Zvlášť důležitá je pro nás proměnná multicastGroups, kam se ukládají multicastové adresy, které jsou na daném rozhraní připojeny. Pro přidání a odebrání multicastových skupin z této proměnné byly vytvořeny funkce:

- addGroupToInterface()
- removeGroupFromInterface ()

Pro další potřebné hodnoty pro IGMP jsme vytvořily vlastní tabulku rozhraní igmpIft. Do této tabulky jsou vkládány jednotlivé záznamy o rozhráních IGMPInterface.

```

struct IGMPInterface
{
    int intID;                // ID rozhrani
    InterfaceEntry *ie;      // ukazatel na info o rozhrani
    bool querier;            // role rozhrani – Querier x Non-Querier
    IPAddress ipQuerier;     // IP adresa Querieru

    // tabulka clenstvi
    std::vector<IGMPGroupMembership> groupMembershipTable;

    // ukazatele na casovace
    IGMPTimer *initGeneralQuery;
};

```

```

    IGMPtimer *generalQuery;
    IGMPtimer *otherQuerierPresent;
};

```

Záznamy o skupinách jsou uloženy v tabulce skupin `groupMembershipTable`, která se pro každé rozhraní vytváří v rámci `IGMPInterface`.

```

struct IGMPGroupMembership
{
    IPAddress multicastGroup;           // adresa mulicastove skupiny
    IPAddress lastReporter;           // adresa Last Reporter
    // ukazatele na casovace
    IGMPtimer *groupMembership;
    IGMPtimer *lastMember;
};

```

Pozici nějaké multicastové skupiny v tabulce určuje funkce:

- `getGroupMembershipPosition()`.

5.6 Zpracování zpráv

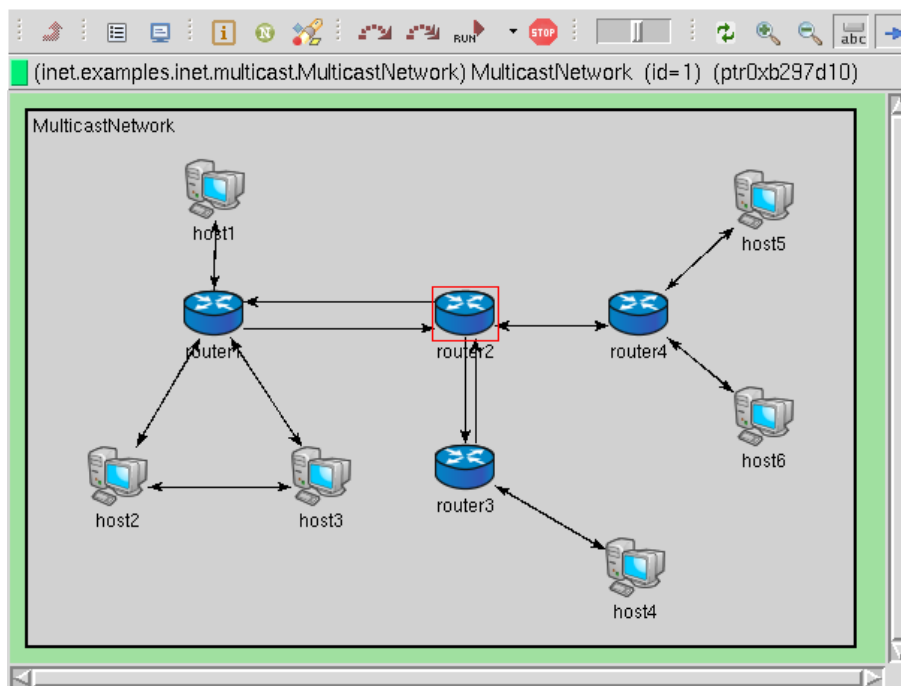
Tak jako v každém modulu OMNeT++, funkce pro přijímání zpráv se jmenuje `handleMessage()`. Ta rozdělí tok zpráv do funkcí `processIGMPtimer()` a `processIGMPmessage()`.

Funkce `processIGMPtimer()` zpracovává příchozí časovače.

Funkce `processIGMPmessage()` zpracovává příchozí zprávy, podle typu je přeposílá příslušným funkcím `processMembershipQuery()`, `processMembershipReportV2()` a `processLeaveGroup()`.

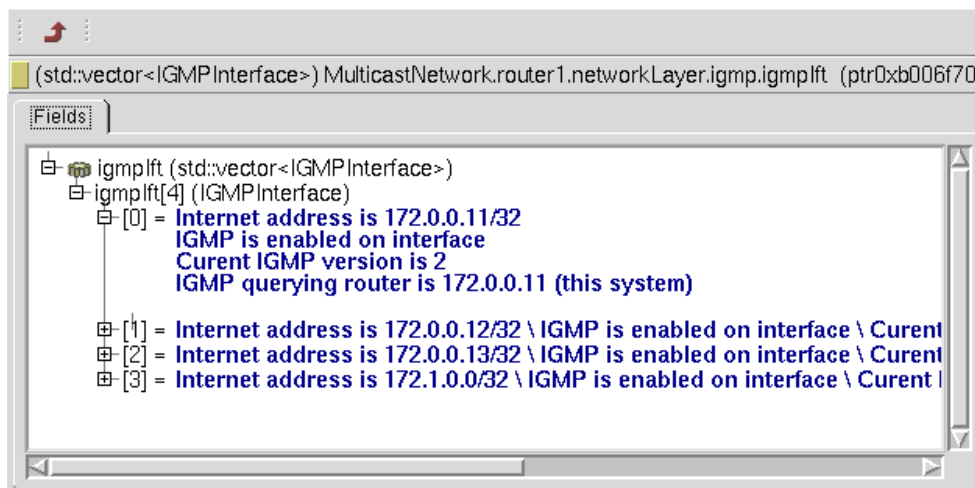
6 Testovací simulace

V této kapitole je popsána testovací simulace, na které jsou ověřeny vlastnosti naimplementovaného modulu. Byla použita příkladová simulace Multicast (Obrázek 6.1).



Obrázek 6.1: Testovací topologie

Ke zobrazení stavu informací o IGMP rozhraní můžeme dostat přes tabulku rozhraní `igmpIft`. Jak je vidět na obrázku 6.2, směrovače jsou ve výchozím stavu v roli *Querier*.



Obrázek 6.2: Zobrazení tabulky IGMP rozhraní

6.1 Test pravidelného odesílání Query

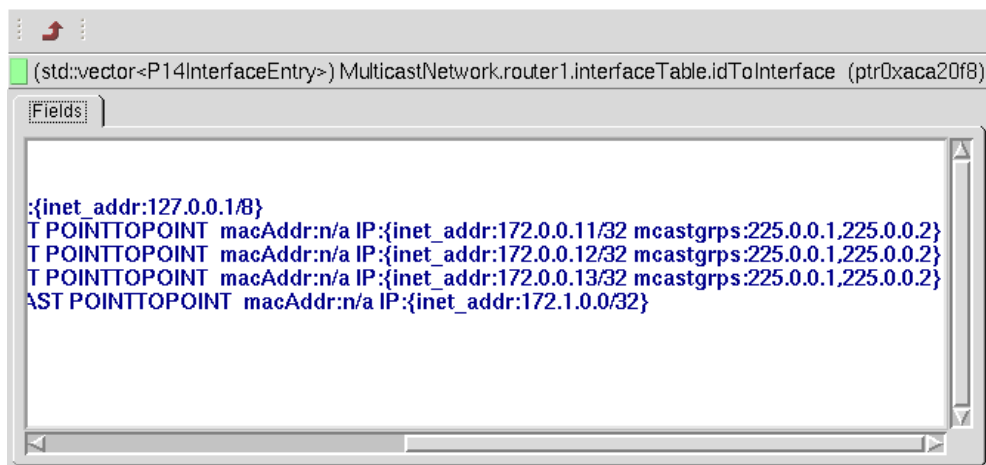
Při inicializaci směrovače začne pravidelné odesílání *Membership Query*. Nejdříve se odešlou inicializační *Membership Query* s intervalem daným časovačem *Timer Init General Query*. Poté se už posílají v intervalu podle časovače *Timer General Query*. Protože byla implementována jen logika směrovače, klientské počítače na tuto zprávu nijak nereagují.

6.2 Test volby dotazovače

Mezi směrovači *router1* a *router2*, *router2* a *router3*, *router2* a *router4* dojde po první obdržené zprávě *Membership Query* k volbě *Querier* směrovače. *Non-Querier* přestane odesílat zprávy *Membership Query*, nastaví se časovač *Timer Other Querier Present*, který je však v naší simulaci neustále obnovován sousedními směrovači, tudíž nedojde k jeho vypršení.

6.3 Test přihlášení do skupiny

Klientské stanice začnou vysílat *Membership Report* na adresu 225.0.0.1 a 225.0.0.2. Po přijetí *Reportu* na rozhraní se objeví v *InterfaceTable* nová multicastová skupina.

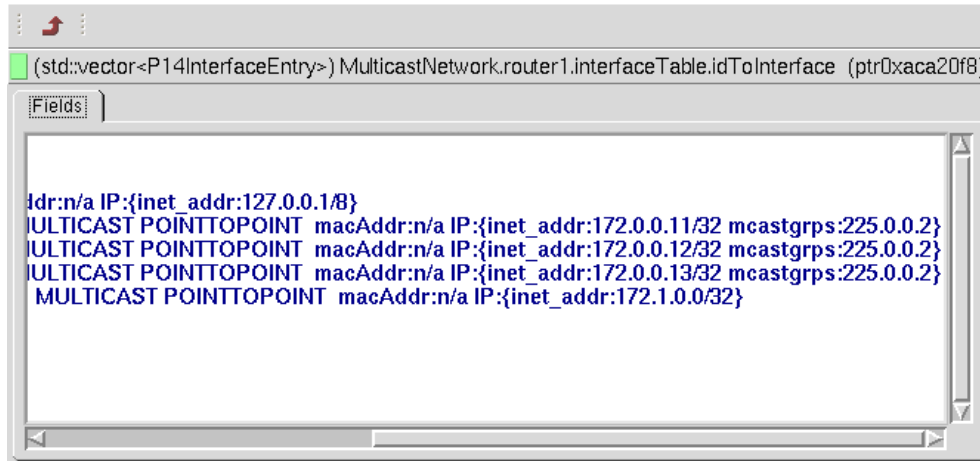


Obrázek 6.3: Rozhraní je zařazeno do multicastových skupin

Pokud klient pošle *Membership Report* na skupinu, která je už na daném rozhraní zařazena, obnoví se jen časovač platnosti skupiny, jinak se změna neprojeví.

6.4 Test odhlášení ze skupiny

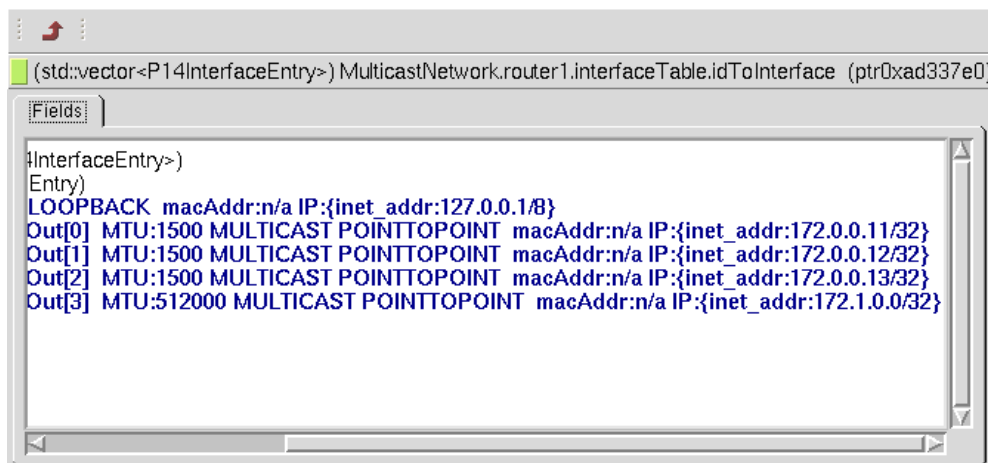
Klient, posílá zprávu *Leave Group* na 225.0.0.1, po jejím obdržení dojde k odstranění skupiny. Zůstane jen skupina 225.0.0.2.



Obrázek 6.4: Stav InterfaceTable po odstranění skupiny.

6.5 Test vypršení platnosti skupiny

Pro skupinu 225.0.0.2 se již žádný *Membership Report* neposílá, ani není požadavek na odhlášení. Skupina zanikne až po vypršení časovače *Timer Group Membership*.



Obrázek 6.5: Stav InterfaceTable po vypršení platnosti skupiny

7 Závěr

Cílem práce bylo popsat multicastovou komunikaci, zjistit možnosti simulace multicasu v OMNeT++ a rozšířit simulátor o zvolený modul.

Vytvoření IGMP modulu pro simulátor OMNeT++ je teprve první krok pro úspěšnou simulaci multicasu. V další fázi je třeba navrhnout a naimplementovat modul protokolu PIM, pro simulaci složitějších modelů bude třeba rozšířit modul přepínače o technologii IGMP snooping. Na těchto rozšířeních se v rámci výzkumné skupiny ANSA v současné době pracuje, proto doufám, že se zanedlouho podaří kompletní multicastovou komunikaci namodelovat.

Další vývoj práce by mohl směřovat v rozšíření IGMP protokolu na nejnovější třetí verzi. V případě budoucího modelování multicasu za použití protokolu IPv6 je nutné implementovat protokol MLD, který má stejný účel jako protokol IGMP. V tomto směru je tedy také možnost dalšího rozvoje této práce.

Během této práce jsem se prostudoval technologii multicasu a porozuměl jejím protokolům. V rámci praktické části jsem se naučil používat simulační nástroj OMNeT++ včetně jeho rozšíření o nový modul.

Literatura

- [1] Veselý, V.: Architektury systémů na Internetu se skupinovým adresováním. Diplomová práce, FIT VUT v Brně, 2009.
- [2] WWW stránky: Internet Multicast Addresses.. [cit. 2010-03-08]
<http://www.iana.org/assignments/multicast-addresses/>
- [3] Deering S.: Host Extensions for IP Multicasting.
<http://www.ietf.org/rfc/rfc1112.txt>, 1989
- [4] Fenner, W.: Internet Group Management Protocol, Version 2.
<http://www.ietf.org/rfc/rfc2236.txt>, 1997
- [5] Cain B., aj.: Internet Group Management Protocol, Version 3.
<http://www.ietf.org/rfc/rfc3376.txt>, 2002
- [6] Deering, S. E.: Host Extensions for IP Multicasting.
<http://tools.ietf.org/html/rfc988>, 1986
- [7] Stewart, B. D., Gough, C.: CCNP BSCI Official Exam Certification Guide, Cisco Press, 2008
- [8] Varga A.: OMNeT++ Discrete Event Simulation System. User Manual, 2008
- [9] WWW stránky: OMNeT++ Community Site. [cit. 2010-05-01]
<http://www.omnetpp.org/>,
- [10] WWW stránky: INET Framework for OMNeT++ 4.0. [cit. 2010-05-01]
<http://inet.omnetpp.org/>

Příloha A

Obsah CD

ansa/ipv4	Upravený modul IP
ansa/igmp	Zdrojové soubory modulu IGMP
ansa/ANSANetworkLayer.ned	Upravený modul NetworkLayer
ansa/ANSARouter.ned	Upravený modul AnsaRouter
ansaExamples/IGMPTest	Testovací simulace
install/	Instalační soubory OMNeT++ a frameworku INET
projekt.pdf	Elektronická verze této práce v PDF
readme.txt	Obsah CD

Příloha B

Implementace IGMP

Hlavičkový soubor IGMP.h

```
// Struktura pro uchovani prislusnosti rozhrani ke skupinam a casovacum.
struct IGMPGroupMembership
{
    IPAddress multicastGroup;    // adresa mulicastove skupiny
    IPAddress lastReporter;     // adresa Last Reporter
    simtime_t startTime;
    IGMPTimer *groupMembership;
    IGMPTimer *lastMember;
};

struct IGMPInterface
{
    int intID;                  // ID rozhrani
    InterfaceEntry *ie;        // ukazatel na popis rozhrani
                                InterfaceEntry
    bool querier;              // querier x non-querier
    IPAddress ipQuerier;       // IP adresa Querier routeru

    std::vector<IGMPGroupMembership> groupMembershipTable;

    /* ukazatele na casovace */
    IGMPTimer *initGeneralQuery;
    IGMPTimer *generalQuery;
    IGMPTimer *otherQuerierPresent;
};

class IGMP : public cSimpleModule
{
private:
    std::vector<IGMPInterface> igmpIft;    //Interní tabulka rozhraní.
```

```

int ROBUSTNESS_VARIABLE;           // default: 2
int QUERY_INTERVAL;                // default: 125 s
int QUERY_RESPONSE_INTERVAL;      // default: 100 = 10 s
int QUERY_RESPONSE_INTERVAL_SEC;
int LAST_MEMBER_QUERY_INTERVAL;   // default: 10 = 1 s
int LAST_MEMBER_QUERY_INTERVAL_SEC;
int LAST_MEMBER_QUERY_COUNT;
int LAST_MEMBER_QUERIER_INTERVAL;
int GROUP_MEMBERSHIP_INTERVAL;
int OTHER_QUERIER_PRESENT_INTERVAL;

// Startup Query Count
// Pocet poslanych zprav General Query pri inicializaci routeru
int STARTUP_QUERY_COUNT;

// Startup Query Interval
// Interval mezi zpravami General Query pri inicializaci routeru
int STARTUP_QUERY_INTERVAL;

// tabulka rozhrani
IInterfaceTable *ift;
int getIGMPInterfaceByID(int interfaceID);

// modul je soucasti modulu routeru
bool isRouter();
IPAddress getQuerierAddress(IPAddress thisRouterAddress, IPAddress
                             otherRouterAddress);

// metody pro praci s casovaci
IGMPTimer * startInterfaceTimer(int timerType, IGMPInterface *
                                igmpInterface);
IGMPTimer * startGroupMembershipTimer(IGMPInterface * igmpInterface,
                                       IPAddress groupAddress, IPAddress lastReporter, int interval);
IGMPTimer * startLastMemberTimer(IGMPInterface * igmpInterface,
                                  IPAddress groupAddress);
void processIGMPTimer(IGMPTimer *timer);

```



```

// tvorba IGMP paketu
IGMPMessage * createPacket(int type, int maxRespTime, IPAddress
                           groupAddress);

// vytvoreni a posilani IGMP zprav
void sendMembershipGeneralQuery(int interfaceID);
void sendMembershipGroupSpecificQuery(IPAddress groupAddress, int
                                       interfaceID);
void sendMembershipReportV2(IPAddress groupAddress, int
                             interfaceID);
void sendLeaveGroup(IPAddress groupAddress, int interfaceID);

// prijem a zpracovani IGMP zprav
void processIGMPMessage(IGMPMessage *igmpmsg);
void processMembershipQuery(IGMPMessage *msg);
void processMembershipReportV2(IGMPMessage *report);
void processLeaveGroup(IGMPMessage *msg);

// sprava clenstvi v multicastovych skupinach na rozhrani
void addGroupToInterface(const int interfaceID, const IPAddress
                         groupAddress);
void removeGroupFromInterface(const int interfaceID, const IPAddress
                              groupAddress);
int getGroupMembershipPosition(IPAddress groupAddress,
                               IGMPInterface * igmpInterface);

protected:
virtual int numInitStages() const {return 5;}
virtual void initialize(int stage);
virtual void handleMessage(cMessage *msg);
};

```